

Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (currently amended) A method for providing a Web service by a plurality of domains through a single IP address, the method comprising:
 - a) For each of the plurality of domains, allocating a server having a unique domain name and the IP address, for providing the service;
 - b) Providing a wrapper, the wrapper being a software module for intermediating between a client of the service and the servers a server providing the service via a standard communication protocol for communicating with the server providing the service each of the servers;
 - c) Upon receiving a request for connecting a client to the server the one of the servers in order to provide the service:
 - (i) Identifying the target domain name of the request by interacting between the client and the wrapper via the standard communication protocol, wherein the target domain name and the user name are embedded in a single command, separated by a symbol that is permitted by the standard communication protocol, wherein the symbol is at least one character that does not conform with the standard characters allowed in a username in the standard communication protocol,

wherein the username phrase is one of “user%domain” and “domain%user”, in which “user” is the username, “domain” is the domain name, and “%” is any symbol that does not conform to the standard communication protocol;

(ii) Mediating the interaction between the wrapper and the server providing the service which is associated with using the target domain name by the standard communication protocol;

(iii) Establishing a communication channel between the server and the client utilizing the standard communication protocol;

(iv) keeping the wrapper active only until the requested server is identified, and the communication is handed to the requested server, and then terminating the wrapper; and

(iv) (v) Allowing the server to provide the service to the client.

2. (canceled)

3. (canceled)

4. (currently amended) The method according to claim 1, wherein the Web services are any of HTTP, FTP, POP3, SMTP, MIRC, Telnet, SSH, Rtelnet, and Shell.

5. (previously presented) The method according to claim 1, wherein each of the domains refers to a different Virtual Dedicated Server.

6. (previously presented) The method according to claim 1, wherein the IP address is associated with a computer system running any dialect of Unix, Solaris, Linux (including Red Hat, Debian, SuSE, FreeBSD), AIX, HP/UX, Tru64, or Irix.

7. (currently amended) The method according to claim 1, wherein each domain has its own instance of the server, the instance being a virtual server.

8. (currently amended) The method according to claim 7, wherein multiple servers providing services to multiple clients for the servers of at least some domains share the same disk space.

9. (previously presented) The method according to claim 8, wherein only one instance of a server resides at a Host, and is referenced by hard links from the domains.

10. (canceled)

11. (canceled)

12. (currently amended) The method according to claim 1, further comprising providing a new replacement shared library including additional functionality compared to an original shared library to which the standard communication protocol refers.

13. (currently amended) The method according to claim 12, wherein the additional functionality of the new replacement shared library is added to the original shared library by hooking.

14. (previously presented) The method according to claim 13, further comprising providing a buffer to each socket, for retaining temporarily the information received from the client, and reading the data from the buffer if it is not empty, or from the socket if the buffer is empty.

15. (previously presented) The method according to claim 14, further comprising ignoring any write command until the buffer is empty.

16. (currently amended) The method according to claim 1, wherein the same a single encryption key is used for all domains on each Host.

17. (previously presented) The method according to claim 1, wherein the wrapper is provided with information related to secured services of the target domain in plain text.

18. (currently amended) A system for providing a Web service to a client by a plurality of domains, through a single IP address, the system comprising:

[-] a plurality of servers, each server providing the service for a corresponding domain;
and

[-] a wrapper, for intermediately between the client and the plurality of servers, using a standard communication protocol,

wherein, for each request to connect for connecting the client to the server, the wrapper identifies a target domain name by interacting with the client via the standard communication protocol, interacts with the server associated with the target domain name via the standard communication protocol, and enables the server to provide the service to the client,

wherein the target domain name and the user name are embedded in a single command, separated by a symbol that is permitted by the standard communication protocol, wherein the symbol is at least one character that does not conform with the standard characters allowed in a username in the standard communication protocol,

wherein the username phrase is one of “user%domain” and “domain%user”, in which “user” is the username, “domain” is the domain name, and “%” is any symbol that does not conform to the standard communication protocol.

19. (canceled)
20. (canceled)
21. (currently amended) The system according to claim 18, further comprising a new replacement shared library including additional functionality compared to an original shared library to which the standard communication protocol refers.

22. (currently amended) The system according to claim 21, wherein the additional functionality of the new replacement shared library is added to the original shared library by hooking.

23. (previously presented) The system according to claim 22, wherein the additional functionality includes retaining temporarily the information received from the client via a socket into a buffer, and reading the data from the buffer if the buffer is not empty, or from the socket if the buffer is empty.

24. (previously presented) The system according to claim 23, further comprising ignoring any write command until the buffer is empty.

25. (currently amended) The system according to claim 18, wherein the same one encryption key is used for all domains on each server.

26. (previously presented) The system according to claim 25, wherein the wrapper is provided with information related to secured services of the target domain in plain text.

27. (currently amended) The system according to claim 18, wherein each domain has its own instance of the server, the instance being a virtual server.

28. (previously presented) The system according to claim 27, wherein servers corresponding to some of the domains share the same disk space.

29. (previously presented) The system according to claim 28, wherein only one instance of some of the servers resides at a Host, and is referenced by hard links from the domains.

30. (currently amended) A wrapper for handling the connection of clients to a plurality of Web domains hosted by a single Host, in which the connection is handled over the

~~standard communication protocol.~~ The system of claim 18, wherein the wrapper ~~providing~~ provides a buffer to each socket for retaining temporarily information received from a client.

31. (currently amended) The ~~wrapper system~~ according to claim 30, ~~wherein the wrapper providing provides~~ servers hosting the Web domain with additional functionality by hooking a ~~new replacement~~ shared library to an original shared library of the standard communication protocol.

32. (currently amended) The ~~wrapper system~~ according to claim 31, wherein, during the connection, “read” commands read the data from the buffer if it is not empty, or the data from the socket, if the buffer is empty.

33. (currently amended) The ~~wrapper system~~ according to claim 30, wherein the connection further comprises ignoring any write command until the buffer is empty.

34. (new) The method according to claim 1, wherein the services are any of HTTP and FTP.